Towards Accurate Image Coding:
Improved Autoregressive Image Generation with Dynamic Vector Quantization

Mengqi Huang[1], Zhendong Mao[1, 2]*, Zhuowei Chen[1], Yongdong Zhang[1, 2]

[1]University of Science and Technology of China, Hefei, China;

[2]Institute of Artificial intelligence, Hefei Comprehensive National Science Center, Hefei, China

{huangmq, chenzw01}@mail.ustc.edu.cn, {zdmao, zhyd73}@ustc.edu.cn

Abstract

Existing vector quantization (VQ) based autoregressive models follow a two-stage generation paradigm that first learns a codebook to encode images as discrete codes, and then completes generation based on the learned codebook. However, they encode fixed-size image regions into fixed-length codes and ignore their naturally different information densities, which results in insufficiency in important regions and redundancy in unimportant ones, and finally degrades the generation quality and speed. Moreover, the fixed-length coding leads to an unnatural raster-scan autoregressive generation. To address the problem, we propose a novel two-stage framework: (1) Dynamic-Quantization VAE (DQ-VAE) which encodes image regions into variable-length codes based on their information densities for an accurate & compact code representation. (2) DQ-Transformer which thereby generates images autoregressively from coarse-grained (smooth regions with fewer codes) to fine-grained (details regions with more codes) by modeling the position and content of codes in each granularity alternately, through a novel stacked-transformer architecture and shared-content, non-shared position input layers designs. Comprehensive experiments on various generation tasks validate our superiorities in both effectiveness and efficiency. Code will be released at https://github.com/CrossmodalGroup/DynamicVectorQuantization.

1. Introduction

The vision community has witnessed the rapid progress of deep generative models, pushing image generation quality to an unprecedented level. As a fundamental task, generating realistic images from arbitrary inputs (*e.g.*, class labels) can empower humans to create rich and diverse visual content and bring numerous real-world applications. Unify-
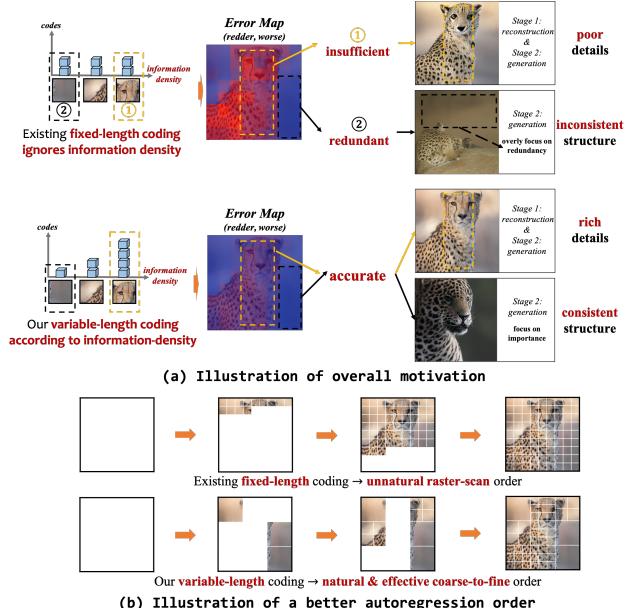

Figure 1. Illustration of our motivation. (a) Existing *fixed-length coding* **ignores information densities**, which results in insufficiency in dense information regions like region ② and redundancy in sparse information regions like region ①, generating poor details and inconsistent structure. Our **information-density-based** *variable-length coding* encodes accurately and produces rich details and consistent structure. (b) Comparison of existing unnatural raster-scan autoregressive generation order and our natural and more effective coarse-to-fine autoregressive generation order. Error map: l_1 loss of each 32^2 region between original images and reconstructions, higher (redder) worse. Existing examples are taken from [13].

ing the realism of local details and the consistency of global structure is the eternal pursuit for all image generations.

Recently, vector quantization (VQ) [37] has been a foundation for various types of generative models as evidenced by numerous large-scale diffusion models like LDM [32], autoregressive models like DALL-E [30], *etc*. These models follow a two-stage generation paradigm, *i.e.*, the first stage learns a codebook by VQ to encode images as dis-

*Zhendong Mao is the corresponding author.

crete codes, where each code represents a local visual pattern, while the second stage learns to generate codes of local regions and then restores to images. The importance lies in that the local details could be well encoded in the first stage and thus the second stage could effectively focus on global structure modeling, leading to better generation quality and scalability. Existing models mainly focus on the second stage to better generate codes for improving generation quality, such as raster-scan autoregression [11, 30, 43], bi-direction [7, 24, 44], or diffusion [5, 14, 32]. Only *a few* works aim to improve the fundamental code representation itself in the first stage, including perceptual and adversarial loss for context-rich codebook [13], residual quantization [23], and more expressive transformer backbone [42], *etc*. Their commonality is that they all focus on encoding more information of all image regions together.

However, existing fundamental encoding works inherently fail to effectively encode image information for an accurate and compact code representation, because they **ignore the naturally different information densities of different image regions** and encode fixed-size regions into *fixed-length* codes. As a result, they suffer from two limitations: (1) insufficient coding for important regions with dense information, which fails to encode all necessary information for faithful reconstruction and therefore degrades the realism of local details in both stages. (2) redundant coding for unimportant ones with sparse information, bringing huge redundant codes that mislead the second stage to focus on the redundancy and therefore significantly hinder the global structure modeling on important ones. As shown in Figure 1(a), the fixed-length codes result in large reconstruction errors in important cheetah regions and produce poor local details (*e.g.*, face, hair) in both stages. Meanwhile, the fixed-length codes are overwhelmed for unimportant background regions, which misleads the second stage to generate redundant background and inconsistent cheetah structure. Moreover, as shown in Figure 1(b), since all regions are encoded into fixed-length codes, there is no way for the second stage to distinguish their varying importance and thus results in an unnatural raster-scan order [13] for existing autoregressive models [11, 23, 30, 42, 43], which fails to consider the image content for an effective generation.

To address this problem, inspired by the classical information coding theorems [18, 33, 34] and their **dynamic coding principle**, we propose **information-density-based** *variable-length coding* for an accurate and compact code representation to improve generation quality and speed. Moreover, we further propose a natural *coarse-to-fine* autoregressive model for a more effective generation. Specifically, we propose a novel two-stage generation framework: (1) *Dynamic-Quantization VAE (DQ-VAE)* which first constructs hierarchical image representations of multiple candidate granularities for each region, and then uses a novel *Dynamic Grained Coding* module to assign the most suitable granularity for each region under the constraint of a proposed *budget loss*, matching the percentage of each granularity to the desired expectation holistically. (2) *DQ-Transformer* which thereby generates images autoregressively from coarse-grained (smooth regions with fewer codes) to fine-grained (details regions with more codes) to more effectively achieve consistent structures. Considering the distribution of different granularities varying, DQ-Transformer models the position and content of codes in each granularity alternately through a novel *stacked-transformer architecture*. To effectively teach the difference between different granularities, we further design *shared-content* and *non-shared-position* input layers.

Our main contributions are summarized as follows:

Conceptual contribution. We point to the inherent insufficiency and redundancy in existing *fixed-length coding* since they **ignore information density**. For the first time, we propose **information-density-based** *variable-length coding* for accurate & compact code representations.

Technical contribution. (1) We propose *DQ-VAE* to dynamically assign variable-length codes to regions based on their different information densities through a novel *Dynamic Grained Coding module* and *budget loss*. (2) We propose *DQ-Transformer* to generate images autoregressively from coarse-grained to fine-grained for the first time, which models the position and content of codes alternately in each granularity by *stacked-transformer architecture* with *shared-content* and *non-shared position* input layers design.

Experimental contribution. Comprehensive experiments on various generations validate our superiority, *e.g.*, we achieve 7.4% quality improvement and faster speed compared to existing state-of-the-art autoregressive model on unconditional generation, and 17.3% quality improvement compared to existing million-level parameters state-of-the-art models on class-conditional generation.

2. Related Works

2.1. Vector Quantization for Image Generation

Existing *VQ-based* models follow a two-stage paradigm that first learns a codebook to encode images into discrete space and then models the underlying distribution in this discrete space. The *VQ-based* paradigm has attracted increasing interest and is adopted by most milestone generative models, such as latent diffusion [32], DALL-E [30], Parti [43], *etc*. Most works focus on the second stage for better learning in the discrete space, such as discrete diffusion [1, 14, 32, 36, 45], bidirection [5, 7, 24, 44], and the most popular raster-scan autoregression [11, 13, 14, 23, 30, 31, 37, 43]. Only a few works aim to improve the fundamental encoding, *e.g.*, VQGAN [13] introduces perceptual and adversarial loss for a context-rich codebook. [23] intro-
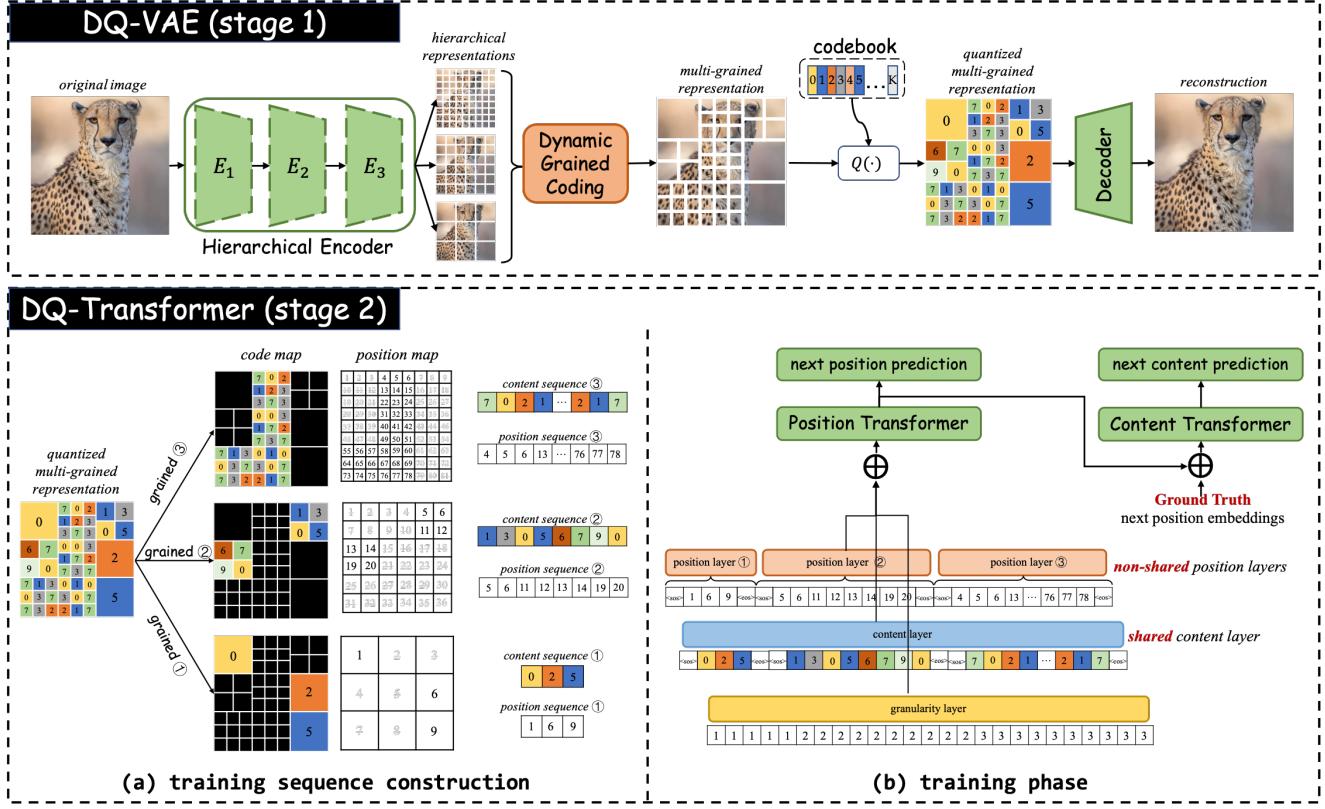
Figure 2. The overview of our proposed two-stage framework. (1) DQ-VAE dynamically assigns variable-length codes for each image region through *Dynamic Grained Coding (DGC)* module. (2) DQ-Transformer models the position and content of codes alternately by the stacked *Position-Transformer* and *Content-Transformer*, generating images autoregressively from coarse-grained to fine-grained. To effectively teach the difference between granularities, we further design *shared-content*, *non-shared-position*, and *granularity* input layers.

duces residual-quantization. [42] proposes a more expressive transformer backbone. Recently, [44] proposes to insert spatially variant information. However, existing fixed-length coding ignores information density and is thus limited by insufficiency and redundancy. For the first time, we propose information-density-based variable-length coding and a more natural coarse-to-fine autoregression.

2.2. Dynamic Network

Designing dynamic architectures is an effective approach for efficient deep learning and yields better representation power and generality [15]. Literately, current research can be mainly categorized into three directions, *i.e.*, dynamic depth for network early exiting [4] or layer skipping [39], dynamic width for skipping neurons [3] or channels [26] and dynamic routing for multi-branch structure networks [17, 25, 35, 41]. Our work belongs in the last direction. To the best of our knowledge, the dynamic network has never been studied in VQ-based generation and we present the first work to realize the variable-length coding of classical information coding theorems through the dynamic network.

3. Methodology

Our overall two-stage framework is depicted in Figure 2. In the following, we will first briefly revisit the formulation of VQ and then describe our proposed method in detail.

3.1. Preliminary

Vector Quantization (VQ) [37] denotes the technique that learns a codebook to encode images into discrete code representations. Formally, the codebook is defined as $\mathcal{C} := \{(k, \boldsymbol{e}(k))\}_{k \in [K]}$, where K is the codebook size and n_z is the dimension of codes. An image $\mathbf{X} \in \mathbb{R}^{H_0 \times W_0 \times 3}$ is first encoded into grid features $\mathbf{Z} = E(\mathbf{X}) \in \mathbb{R}^{H \times W \times n_z}$ by the encoder E, where $(H, W) = (H_0/f, W_0/f)$ and f is the downsampling factor. For each vector $\boldsymbol{z} \in \mathbb{R}^{n_z}$ in $\mathbf{Z}$, the quantization operation $\mathcal{Q}(\cdot)$ replaces it with the code that has the closest euclidean distance with it in the codebook $\mathcal{C}$:

$$\mathcal{Q}(\boldsymbol{z}; \mathcal{C}) = \arg \min_{k \in [K]} ||\boldsymbol{z} - \boldsymbol{e}_k||_2^2. \quad (1)$$

Here, $\mathcal{Q}(\boldsymbol{z}; \mathcal{C})$ is the quantized code. $\boldsymbol{z^q} = \boldsymbol{e}(\mathcal{Q}(\boldsymbol{z}; \mathcal{C}))$ is the quantized vector. Therefore, the quantized encoded fea-

tures are $\mathbf{Z}^q \in \mathbb{R}^{H \times W \times n_z}$. The decoder D is used to reconstruct the original image by $\tilde{\mathbf{X}} = D(\mathbf{Z}^q)$. Here each code roughly represents a fixed f^2 size visual pattern and each image region is represented by the same length of codes without distinguishing their different information densities. As a result, existing works suffer from both insufficiency in important regions and redundancy in unimportant ones.

3.2. Stage 1:Dynamic-Quantization VAE(DQ-VAE)

Different from existing works that adopt a fixed downsampling factor f to represent image regions as fixed-length codes, DQ-VAE first defines a set of candidates:

$$\mathrm{F} = \{f_1, f_2, ..., f_K\}, \text{where } f_1 < f_2 < ... < f_K, \quad (2)$$

and encodes images into hierarchical features $\mathbf{Z} = \{\mathbf{Z}_1, \mathbf{Z}_2, ..., \mathbf{Z}_K\}$ through a hierarchical encoder E_h, where $\mathbf{Z}_i \in \mathbb{R}^{H_i \times W_i \times n_z}$ and $(H_i, W_i) = (H_0/f_i, W_0/f_i)$ for each $i \in \{1, 2, ..., K\}$. The image region's size is set as the maximum downsampling factor, i.e., $S = f_K$, and therefore each S^2 size image region now has multiple granularity representations containing different numbers of features. Then the *Dynamic Grained Coding (DGC)* module assigns the most suitable granularity for each region and results in multi-grained representations, which are further quantized by VQ. To deal with the irregular code map that different regions have different numbers of codes, we further propose a simple but effective nearest-neighbor replication, that is, in each region the quantized codes are replicated to the code number of the finest granularity if the finest granularity is not assigned for it, resulting in a regular code map that could be conveniently decoded by the convolutional decoder D.

Dynamic Grained Coding (DGC) module. As illustrated in Figure 3, given the encoded hierarchical image features $\mathbf{Z} = \{\mathbf{Z}_1, \mathbf{Z}_2, ..., \mathbf{Z}_K\}$, we implement a discrete gating network with Gumbel-Softmax technique [19] to determine the granularity for each image region. Specifically, each granularity feature is first normed by group-normalization to stabilize training and then pooled to the size of the coarsest granularity feature by average-pooling, except the coarsest granularity (i.e., f_K) feature itself. The pooled features are denoted as $\mathbf{Z}' = \{\mathbf{Z}'_1, \mathbf{Z}'_2, ..., \mathbf{Z}'_K\}$ and $\mathbf{Z}'_i \in \mathbb{R}^{H_s \times W_s \times n_z}$ for $i \in \{1, 2, ..., K\}$, where $(H_s, W_s) = (H_0/f_K, W_0/f_K)$. The gating logits $\mathbf{G}$ are generated as:

$$\mathbf{G} = (\mathbf{Z}'_1 \| \mathbf{Z}'_2 \| ... \| \mathbf{Z}'_K)\mathbf{W_g} \in \mathbb{R}^{H_s \times W_s \times k}, \quad (3)$$

where $\|$ is the concatenation operation along the channel dimension and $\mathbf{W_g} \in \mathbb{R}^{(K \times n_z) \times K}$ is the learnable weight. For each region (i, j), its gating logits $g_{i,j} \in \mathbb{R}^K$ is used to decide the granularity by calculating the gating index:

$$\theta_{i,j} = \arg \max_k (g_{i,j,k}) \in \{1, 2, ..., K\}. \quad (4)$$
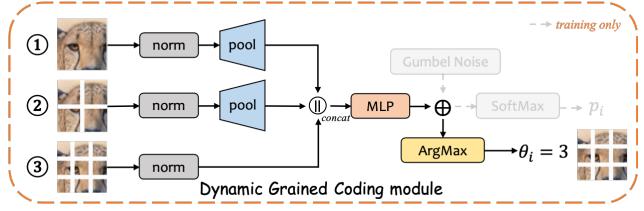

Figure 3. Illustration of our Dynamic Grained Coding module.

To enable the end-to-end training of this discrete process, inspired by [40, 46], the determined decisions in Eq.(4) are replaced with the stochastic sampling process. Mathematically, given a categorical distribution with unnormalized log probabilities, discrete gating indices can be yielded with noise samples drawn from a standard Gumbel distribution:

$$\theta_{i,j} = \arg \max_k (g_{i,j,k} + n_k), \text{where } n_k \sim \mathrm{Gumbel}(0,1). \quad (5)$$

To enable the back-propagation of the above hard decision, we adopt the Gumbel-Softmax technique [19] to give a continuous and differentiable approximation by replacing the argmax with a softmax operation. The soft gating score $p_{i,j}$ for each region is then selected by the gating indices:

$$p_{i,j} = \frac{\exp((g_{i,j,\theta_{i,j}} + n_{\theta_{i,j}}))/\tau}{\sum_k^K \exp((g_{i,j,k} + n_k)/\tau)} \in [0, 1], \quad (6)$$

where the temperature $\tau = 1$. We use a straight-through estimator for the gradients of gating logits, which are obtained through the soft gating score $p_{i,j}$ during the backward pass. The above stochastic process is only adopted during training and no random sampling is required during inference.

Budget Loss. We adopt the training loss of VQGAN [13] as $\mathcal{L}_{vanilla}$, which includes reconstruction loss (l_1 loss, perceptual loss, adversarial loss) and quantization loss. In the absence of a budget constraint, the DGC module typically prefers to assign the finest granularity for all image regions, which is in contrast to our purpose. Therefore, we further propose a *budget loss* to match the percentage of each granularity to our desired expectation. Specifically, we denote the desired ratio of each granularity k as r_k and $\sum_k^K r_k = 1$. For an image sample whose current assigned ratio of each granularity k is r'_k, we define *budget loss* as:

$$\mathcal{L}_{budget} = \sum_k^{K-1} (r_k - r'_k)^2, \quad (7)$$

where we only calculate on $K - 1$ granularities since the ratio of the last granularity is determined by $1 - \sum_k^{K-1} r_k$. The final loss for DQ-VAE is defined as:

$$\mathcal{L}_{stage1} = \mathcal{L}_{vanilla} + \lambda \mathcal{L}_{budget}, \quad (8)$$

where λ is a loss balance hyper-parameter. The expected ratio of each granularity is holistic on the dataset level. Therefore, since important regions contribute more to the reconstruction quality, the variable-length coding is realized from two aspects, *i.e.*, *inter-dynamic*, longer code sequence for complex images while shorter code sequence for easy ones; *intra-dynamic*, for each image, more codes for important regions while fewer codes for unimportant ones.

3.3. Stage 2: DQ-Transformer

Different images share different perceptually important regions and different complexities. Therefore, DQ-VAE encodes images as the code sequence of variable lengths and the distribution of each granularity region in images is also completely different. Though learning this dynamic underlying prior is very challenging, it also opens a promising potential for autoregressive image generation, that is, a natural and more effective coarse-grained to fine-grained generation order since DQ-VAE naturally divides coarse regions (smooth regions with fewer codes) apart from fine regions (details regions with more codes). Imagine image generation as a jigsaw puzzle problem, it is more effective and efficient that we first fill in the large and easy pieces (coarse regions) and then fill in the small and difficult ones (fine regions). With this motivation, DQ-Transformer first constructs the codes' content and position sequence in each granularity separately and then concatenates them in a coarse-to-fine manner to autoregressively predict the next code's position and content through the stacked *Position-Transformer* and *Content-Transformer*. The distinction of different granularities is realized by the *shared content*, *non-shared-position*, and *granularity* input layers designs.

Training sequence construction. As illustrated in stage 2(a) in Figure 2, the sequence of each granularity is constructed separately. As for the content sequence, each index is the quantized code index. As for the position sequence, each index is the position of the corresponding code index in the *position map of current granularity*. We add a special <sos> code at the beginning of all content and position sequences to indicate the start of the sequence, and another special <eos> code at the end of them to indicate the end of the sequence. To enable batch training and sampling, we use a special <pad> code to pad all samples to the same length in each granularity. Finally, we concatenate all granularities' content and position sequences in a coarse-to-fine manner, which we denote as C and P, respectively.

Position-Transformer. We first learn to predict the next code position conditioned on all previous codes and their positions. The input of Position-Transformer consists of four parts: (1) content embedding which is calculated from C by a *shared-content* layer for all granularities, (2) position embedding which is calculated from P by *non-shared-position* layers for each granularity separately, (3) granular-

ity embedding which is used for distinguishing each granularity, and (4) a learned absolute position embedding for making the network aware of the absolute position of the sequence, which is the same as most transformer-architecture [13, 36, 38]. After processing by Position-Transformer, the output hidden vector H_p encodes both code and their position information and is used for next position predicting. The negative log-likelihood (NLL) loss for the next code position autoregressive training is:

$$\mathcal{L}_{position} = \mathbb{E}(-\log p(P_l | P_{<l}, C_{<l})) \qquad (9)$$

Content-Transformer. We then learn to predict the next code's content conditioned on all previous codes and the position of *current* code. Specifically, The input of Content-Transformer is two parts: (1) the output of Position-Transformer H_p and (2) the ground-truth information of the current position which also is calculated by the *non-shared-position* layers. For example, if the input position sequence for Position-Transformer is $P_{[0:-2]}$, then the input ground-truth position sequence for Content-Transformer is $P_{[1:-1]}$. The negative log-likelihood (NLL) loss for the next code's content autoregressive training is:

$$\mathcal{L}_{content} = \mathbb{E}(-\log p(C_l | P_{\leq l}, C_{<l})) \qquad (10)$$

Training & **Inference.** During training, the total loss for DQ-Transformer is defined as:

$$\mathcal{L}_{stage2} = \mathcal{L}_{position} + \mathcal{L}_{content}. \qquad (11)$$

Our proposed DQ-Transformer is a general visual generative model which could be easily extended to various other generation tasks. As for the class-conditional generation, we replace the <sos> code in the content sequence of each granularity with the class-label code. During inference, we could also autoregressively generate images from coarse-grained to fine-grained, as illustrated in Algorithm 1, where we take the unconditional generation as an example and other conditional generations can be derived accordingly.

4. Experiments

Benchmarks. We evaluate our method on unconditional FFHQ [20] benchmark and class-conditional ImageNet [9] benchmark with 256×256 image resolution.

Metrics. Following previous works [13,23,44], the standard Fréchet Inception Distance (FID) [16] is adopted for evaluating the generation and reconstruction quality (denoted as rFID). rFID is calculated over the entire test set. Inception Score (IS) [16] is also adopted for class-conditional generation on the ImageNet benchmark.

Implementation. DQ-VAE follows the architecture of VQGAN [13] except for the lightweight DRC module, which is trained with the codebook size $K = 1024$ and $\lambda =$

Figure 4. Qualitative results. Left: Our unconditional generation on FFHQ. Right: Our class-conditional generation on ImageNet.

Algorithm 1 Unconditional batch sampling.

Input: The granularity number K and batch size B;
    The initial empty position (code) sequence P (C).
Output: The generated image $\mathcal{I}$.
1: **for** each $k \in [1, K]$ **do**
2:     // sample each granularity in a coarse-to-fine order
3:     $P = \text{concat}(P, <\text{sos}>)$, $C = \text{concat}(C, <\text{sos}>)$
4:     **while** NOT all samples have sampled $<\text{eos}>$ **do**
5:         mask sampled position indexes to avoid repeat
6:         sample next code position $P_i \in \mathbb{R}^B$
7:         **if** $P_{i,b} == <\text{eos}>$, for $b \in [1, B]$ **then**
8:             $P_{>i,b} = <\text{pad}>$
9:             // if sampled $<\text{eos}>$, the following will only can be
               $<\text{pad}>$ for this sample in current granularity
10:         **end if**
11:         sample next code C_i
12:         $C = \text{concat}(C, C_i)$, $P = \text{concat}(P, P_i)$
13:     **end while**
14: **end for**
15: **return** decoded image $\mathcal{I}$ from P and C

---

10. DQ-Transformer adopts a stack of causal self-attention blocks [38] and is trained with two different settings, *i.e.*, DQ-Transformer$_b$(base) with 6 layers Position-Transformer and 18 layers Content-Transformer of a total 308M parameters, and DQ-Transformer$_l$(large) with 6 layers Position-Transformer and 42 layers Content-Transformer of a total 608M parameters to demonstrate our scalability. All models are trained with eight RTX-3090 GPUs. Top-k and top-p sampling are used to report the best performance. More details can be found in the supplementary.

4.1. Comparison with state-of-the-art methods

The main results are reported on dual granularities of F = $\{8, 16\}$, and the ratio $r_{f=8} = 0.5$ (640 average length).

Unconditional generation. As shown in Table 1, our model outperform all existing autoregressive state-of-the-art models including the strongest large-scale ViT-VQGAN [42] by a 7.4% quality improvement. We compare with other types of state-of-the-art models in Table 4 and also achieve top-level performance. The qualitative results of unconditional generation are shown on the left of Figure 4.

Class-conditional generation. The comparison is split

Model	L	#Params	FID↓
VQGAN$_{('21)}$ [13]	256	(72.1+307)M	11.4
DCT$_{('21)}$ [28]	>1024	738M	13.06
ViT-VQGAN$_{('22)}$ [42]	1024	(599+1697)M	5.3
RQ-VAE$_{('22)}$ [23]	256	(100+355)M	10.38
Mo-VQGAN$_{('22)}$) [44]	1024	(82.7+307)M	8.52
DQ-Transformer$_b$	640	**(47.5+308)M**	**4.91**

Table 1. Comparison of unconditional autoregressive generation on FFHQ. L is coding length. #Params splits in (VAE+autoregressive model).

Type	Model	L	#Params	FID↓	IS↑
GAN	BigGAN-deep [6]	-	160M	6.95	198.2
diffusion	[29]	-	280M	12.26	-
diffusion	ADM [10]	-	554M	10.94	101.0
bi-direct	MaskGIT [7]	1024	227M	6.18	182.1
ARM	VQGAN* [13]	256	379M	17.5	75
ARM	DCT [28]	>1024	738M	36.5	-
ARM	RQ-VAE [23]	256	480M	15.72	86.8
ARM	RQ-VAE [23]	256	821M	13.11	104.3
ARM	Mo-VQGAN [44]	1024	389M	7.12	138.3
ARM	DQ-Transformer$_b$	640	355M	7.34	152.8
ARM	**DQ-Transformer$_l$**	640	655M	**5.11**	**178.2**

Table 2. Comparison of class-conditional generation with million-level parameters on ImageNet. L is coding length. ARM denotes for autoregressive model. * denotes for our reproduction.

into Million/Billion according to whether they can be trained under normal computing resources (*i.e.*, 24G memory 3090). We first compare with all million-level parameters state-of-the-art in Table 2. Our model with 355M parameters already outperforms all autoregressive and diffusion models. Moreover, our model with 655M outperforms GAN-based and bi-direct state-of-the-art, which demonstrates our effectiveness and scalability. We further compare with large-scale billion-level state-of-the-art in Table 3, where we achieve top-level performance with fewer parameters. The qualitative results of class-conditional generation are shown on the right of Figure 4.

4.2. Ablations & Analysis

Analysis on DQ-VAE. We first demonstrate that our variable-length coding has better reconstruction compared to the existing fixed-length one in Table 5. We take VQ-
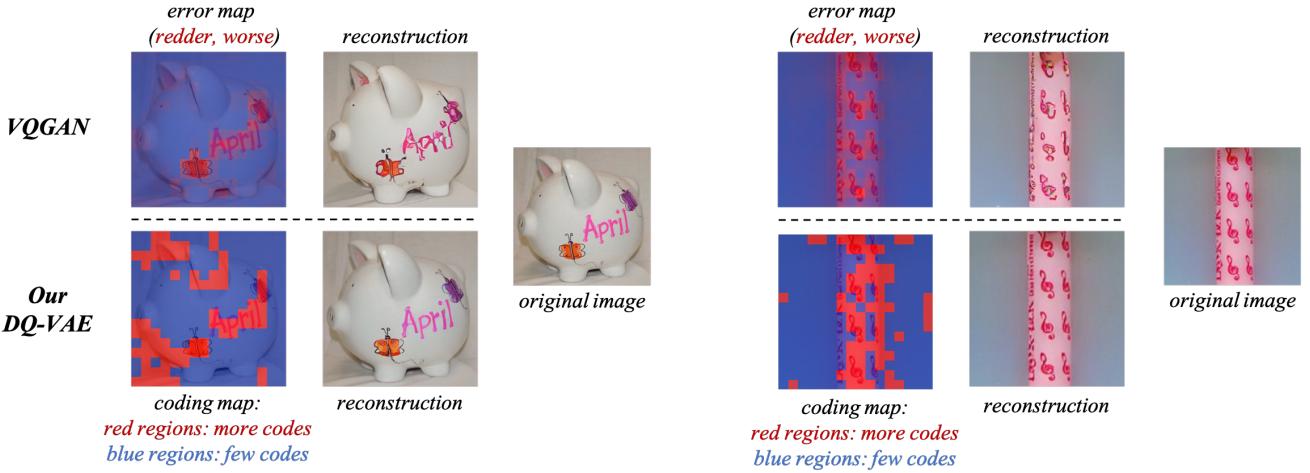
Figure 5. Visualization of the variable-length coding of our DQ-VAE, where **our coding map exactly matches the error map of VQGAN** and therefore leads to better reconstruction quality, *i.e.*, the **information-dense regions** where VQGAN has **higher reconstruction error** are assigned to **more codes**, while **information-sparse regions** where VQGAN has **lower reconstruction error** are assigned to **few codes**.

Type	Model	L	#Params	FID↓	IS↑
Diffusion	ImageBART [12]	-	3.5B	21.19	61.6
ARM	VQ-VAE-2 [31]	5120	13.5B	31.11	45
ARM	VQGAN [13]	256	1.4B	15.78	78.3
ARM	ViT-VQGAN [36]	1024	2.2B	4.17	175.1
ARM	RQ-VAE [23]	256	3.8B	7.55	134
ARM	DQ-Transformer$_b$	640	355M	7.34	152.8
ARM	**DQ-Transformer$_l$**	640	**655**M	**5.11**	**178.2**

Table 3. Comparison between **our million-level model** and **large-scale billion-level big models** of class-conditional generation on ImageNet.

Model Type	Model	FID↓
GAN	BigGAN [6]	12.4
GAN	StyleGAN2 [21]	3.8
VAE	VDVAE [8]	28.5
Diffusion	ImageBART [12]	9.57
Diffusion	UDM [22]	5.54
Autoregressive	**DQ-Transformer$_b$**	**4.91**

Table 4. Comparison with other types of state-of-the-art on unconditional FFHQ, where we further improve the quality of autoregressive models.

Model	F	ratio	rFID↓
VQGAN [13]	16	-	4.82
DQ-VAE	{8,16,32}	{0.05, 0.75, 0.3}	4.57
DQ-VAE	{8,16,32}	{0.075, 0.625, 0.3}	4.08
DQ-VAE / random	{8,16,32}	{0.075, 0.625, 0.3}	7.32
DQ-VAE	{8,16,32}	{0.1, 0.5, 0.4}	4.96
DQ-VAE	{8,16,32}	{0.125, 0.375, 0.5}	6.39

Table 5. Ablations of the proposed variable-length coding on ImageNet. Here F denotes the granularity candidates set. "ratio" denotes the ratio of each granularity. We show that variable-length coding could bring better reconstruction compared to fixed-length coding on the same code length.

GAN [13] of $f = 16$ as the baseline, and DQ-VAE adopts triple granularities of F = $\{8, 16, 32\}$ and subject to:

$$r_{f=32} = 4 \times r_{f=8}, \tag{12}$$

which ensures DQ-VAE's expected mean code length is the same as VQGAN (*i.e.*, 256). We could conclude: (1) With a proper ratio, DQ-VAE's variable-length coding achieves better reconstruction quality compared to VQGAN's fixed-length one (ours 4.08 *vs.* VQGAN's 4.82). The reason is that important regions require more codes to encode necessary information, while fewer codes are enough for unimportant ones since they are less informative. The phenomenon also reveals that existing fixed-length coding is both insufficient in important regions and redundant in unimportant ones. (2) When we improperly increase $r_{f=8}$, we get a larger $r_{f=32}$ which will inevitably assign some important regions with fewer codes and thus degrade the reconstruction quality. (3) Moreover, DQ-VAE's adaptive assignment significantly outperforms the random one (ours 4.08 *vs.* random's 7.32) which demonstrates that DQ-VAE could distinguish important regions from unimportant ones.

We then analyze the impact of different ratio percentages in Table 6, where DQ-VAE adopts dual granularities of F = $\{8, 16\}$. We show that: (1) The mean code length of each ratio matches the expectation well, which validates our proposed budget loss. (2) The results are consistent with *the Pareto principle*, which is also known as *20/80 laws*. To be specific, when increasing $r_{f=8}$ from 0 to 0.3, we get 1.44 FID improvement while only a slight codebook usage drop, which indicates that the first 30% percentage important regions contribute the most valid information of images and existing fixed-length coding is insufficient in them. Mean-

Model	$r_{f=8}$	mean (expected)	var	rFID↓	usage ↑
VQGAN	0	256	-	4.46	63.89%
DQ-VAE	0.1	332 (333)	760.6	3.6	63.02%
DQ-VAE	0.3	494 (486)	621.3	3.02	62.3%
DQ-VAE	0.5	646 (640)	348.3	2.38	59.9%
DQ-VAE	0.7	792 (794)	285.6	2.09	56.01%
DQ-VAE	0.9	945 (947)	132.4	1.87	52.32%
VQGAN	1	1024	-	1.8	46.41%

Table 6. Ablations of different granularity ratios of DQ-VAE with F={8, 16} on FFHQ. Here $r_{f=8}$ denotes the ratio of $f = 8$. "mean" and "var" denote the mean and variance of dynamic coding length. The codebook usage is calculated as the percentage of used codes over the entire test set.

Content	Position	Granularity	Absolute position	FID↓
shared	non-shared	✓	✓	4.91
non-shared	non-shared	✓	✓	5.54
shared	shared	✓	✓	18.28
shared	non-shared	✗	✓	16.87
shared	non-shared	✓	✗	5.06

Table 7. Ablations of DQ-Transformer input designs on FFHQ. Here "granularity" denotes for DQ-Transformer's granularity layer.
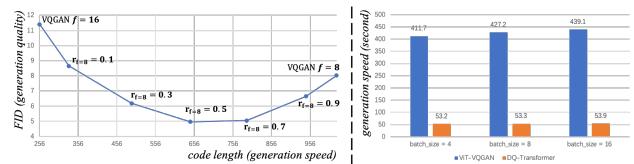

Figure 6. Left: The Pareto curves of the different ratios between generation quality (FID) and generation speed (code length) on FFHQ. Right: The speed comparison between large-scale ViT-VQGAN [42] and our DQ-Transformer(base) according to different batch sizes on FFHQ.

while, when increasing $r_{f=8}$ from 0.7 to 1.0, we only get a subtle 0.21 FID improvement but a significant 9.6% codebook usage drop, which indicates that the last 30% percentage unimportant regions contribute little valid information of images but most redundancy. The experimental results strongly support our motivations for variable-length coding to get rid of insufficiency and redundancy simultaneously.

We visualize our variable-length coding on ImageNet in Figure 5, where DQ-VAE adopts dual granularities of F = {8, 16} and $r_{f=8} = 0.3$. The error map is calculated by l_1 loss of each 16^2 size region between images and VQGAN ($f = 16$) reconstructions. The red regions in our coding map are assigned to $f = 8$ (4 codes) while the blue ones are assigned to $f = 16$ (1 code). We show that our coding map matches VQGAN's error map, *i.e.*, important regions are assigned to more codes and unimportant ones are assigned to few codes, leading to better reconstruction quality.

Analysis on the effectiveness of DQ-Transformer. We first validate our input layers designs in Table 7. The *non-shared-position* and *granularity* layers are very important since they distinguish different granularities. Without these designs, DQ-Transformer fails to know which granularity of code should be generated next, and thus performs worse.

We then analyze the generation quality of different ratios in Figure 6 left. The generation speed of autoregressive models mostly depends on their code length. The Pareto curve shows that the generation quality (FID) saturates when $r_{f=8}$ reaches 0.5. The experimental phenomenon reveals that a proper ratio is important for the unity of a high generation quality and fast generation speed since it guarantees effective coding in both important regions and unim-

portant ones for an accurate & compact code representation.

Analysis on the efficiency of DQ-Transformer. We compare our generation speed to the existing state-of-the-art autoregressive model ViT-VQGAN [42] according to different batch sizes in Figure 6 right. The generation speeds are evaluated on a single RTX-3090 GPU and the setup of ViT-VQGAN is implemented the same as its original paper. Our model achieves a much faster generation speed for all batch sizes which validates the efficiency brought by our accurate and compact code representation.

5. Conclusion & Future Direction

In this study, we point out that the existing fixed-length coding ignores the naturally different information densities of image regions and is inherently limited by insufficiency and redundancy, which degrades generation quality and speed. Moreover, the fixed-length coding brings an unnatural raster-scan autoregression. We thereby propose a novel two-stage generation framework: (1) *DQ-VAE* which dynamically assigns variable-length codes to regions based on their information densities for an accurate and compact code representation. (2) *DQ-Transformer* which then models the position and content of codes alternately, generating images autoregressively in a more natural and effective coarse-to-fine order for the first time. To effectively teach the difference between different granularities, we further design *shared-content*, *non-shared-position*, and *granularity* input layers. Comprehensive experiments on various image generations validate our effectiveness and efficiency.

Future Direction. VQ is the foundation for modern autoregressive [11, 23, 30, 43, 44], discrete diffusion [14, 32], and bidirectional [7] generation, and even pretraining [2, 27]. Our study validates the effectiveness and efficiency of the variable-length coding for autoregressive generation, but its great potential for diffusion, bi-direction, and pretraining is worth further exploration in the future.

6. Acknowledgments

References

[1] Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34:17981–17993, 2021. 2

[2] Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021. 8

[3] Emmanuel Bengio, Pierre-Luc Bacon, Joelle Pineau, and Doina Precup. Conditional computation in neural networks for faster models. *arXiv preprint arXiv:1511.06297*, 2015. 3

[4] Tolga Bolukbasi, Joseph Wang, Ofer Dekel, and Venkatesh Saligrama. Adaptive neural networks for efficient inference. In *International Conference on Machine Learning*, pages 527–536. PMLR, 2017. 3

[5] Sam Bond-Taylor, Peter Hessey, Hiroshi Sasaki, Toby P Breckon, and Chris G Willcocks. Unleashing transformers: Parallel token prediction with discrete absorbing diffusion for fast high-resolution image generation from vector-quantized codes. *arXiv preprint arXiv:2111.12701*, 2021. 2

[6] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018. 6, 7

[7] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11315–11325, 2022. 2, 6, 8

[8] Rewon Child. Very deep vaes generalize autoregressive models and can outperform them on images. *arXiv preprint arXiv:2011.10650*, 2020. 7

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 5

[10] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021. 6

[11] Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, et al. Cogview: Mastering text-to-image generation via transformers. *Advances in Neural Information Processing Systems*, 34:19822–19835, 2021. 2, 8

[12] Patrick Esser, Robin Rombach, Andreas Blattmann, and Bjorn Ommer. Imagebart: Bidirectional context with multinomial diffusion for autoregressive image synthesis. *Advances in Neural Information Processing Systems*, 34:3518–3532, 2021. 7

[13] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021. 1, 2, 4, 5, 6, 7

[14] Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10696–10706, 2022. 2, 8

[15] Yizeng Han, Gao Huang, Shiji Song, Le Yang, Honghui Wang, and Yulin Wang. Dynamic neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 3

[16] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 5

[17] Mengqi Huang, Zhendong Mao, Penghui Wang, Quan Wang, and Yongdong Zhang. Dse-gan: Dynamic semantic evolution generative adversarial network for text-to-image generation. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 4345–4354, 2022. 3

[18] David A Huffman. A method for the construction of minimum-redundancy codes. *Proceedings of the IRE*, 40(9):1098–1101, 1952. 2

[19] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016. 4

[20] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 5

[21] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020. 7

[22] Dongjun Kim, Seungjae Shin, Kyungwoo Song, Wanmo Kang, and Il-Chul Moon. Score matching model for unbounded data score. *arXiv preprint arXiv:2106.05527*, 2021. 7

[23] Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11523–11532, 2022. 2, 5, 6, 7, 8

[24] Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Draft-and-revise: Effective image generation with contextual rq-transformer. *arXiv preprint arXiv:2206.04452*, 2022. 2

[25] Yanwei Li, Lin Song, Yukang Chen, Zeming Li, Xiangyu Zhang, Xingang Wang, and Jian Sun. Learning dynamic routing for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8553–8562, 2020. 3

[26] Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning efficient convolutional networks through network slimming. In *Proceedings of the IEEE international conference on computer vision*, pages 2736–2744, 2017. 3

[27] Chengzhi Mao, Lu Jiang, Mostafa Dehghani, Carl Vondrick, Rahul Sukthankar, and Irfan Essa. Discrete representations

strengthen vision transformer robustness. *arXiv preprint arXiv:2111.10493*, 2021. 8

[28] Charlie Nash, Jacob Menick, Sander Dieleman, and Peter W Battaglia. Generating images with sparse representations. *arXiv preprint arXiv:2103.03841*, 2021. 6

[29] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021. 6

[30] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021. 1, 2, 8

[31] Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32, 2019. 2, 7

[32] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 1, 2, 8

[33] Claude Elwood Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948. 2

[34] Claude E Shannon et al. Coding theorems for a discrete source with a fidelity criterion. *IRE Nat. Conv. Rec*, 4(142-163):1, 1959. 2

[35] Lin Song, Songyang Zhang, Songtao Liu, Zeming Li, Xuming He, Hongbin Sun, Jian Sun, and Nanning Zheng. Dynamic grained encoder for vision transformers. *Advances in Neural Information Processing Systems*, 34:5770–5783, 2021. 3

[36] Zhicong Tang, Shuyang Gu, Jianmin Bao, Dong Chen, and Fang Wen. Improved vector quantized diffusion models. *arXiv preprint arXiv:2205.16007*, 2022. 2, 5, 7

[37] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017. 1, 2, 3

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 5, 6

[39] Andreas Veit and Serge Belongie. Convolutional networks with adaptive inference graphs. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–18, 2018. 3

[40] Zhenda Xie, Zheng Zhang, Xizhou Zhu, Gao Huang, and Stephen Lin. Spatially adaptive inference with stochastic feature sampling and interpolation. In *European conference on computer vision*, pages 531–548. Springer, 2020. 4

[41] Le Yang, Yizeng Han, Xi Chen, Shiji Song, Jifeng Dai, and Gao Huang. Resolution adaptive networks for efficient inference. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2369–2378, 2020. 3

[42] Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. Vector-quantized image modeling with improved vqgan. *arXiv preprint arXiv:2110.04627*, 2021. 2, 3, 6, 8

[43] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2022. 2, 8

[44] Chuanxia Zheng, Long Tung Vuong, Jianfei Cai, and Dinh Phung. Movq: Modulating quantized vectors for high-fidelity image generation. *arXiv preprint arXiv:2209.09002*, 2022. 2, 3, 5, 6, 8

[45] Ye Zhu, Yu Wu, Kyle Olszewski, Jian Ren, Sergey Tulyakov, and Yan Yan. Discrete contrastive diffusion for cross-modal and conditional generation. *arXiv preprint arXiv:2206.07771*, 2022. 2

[46] Yichen Zhu, Yuqin Zhu, Jie Du, Yi Wang, Zhicai Ou, Feifei Feng, and Jian Tang. Make a long image short: Adaptive token length for vision transformers. *arXiv preprint arXiv:2112.01686*, 2021. 4